

Trabalho 1 - Processamento de Imagens

Décio Gonçalves de Aguiar Neto¹

¹Universidade Estadual de Campinas - UNICAMP
Instituto de Computação

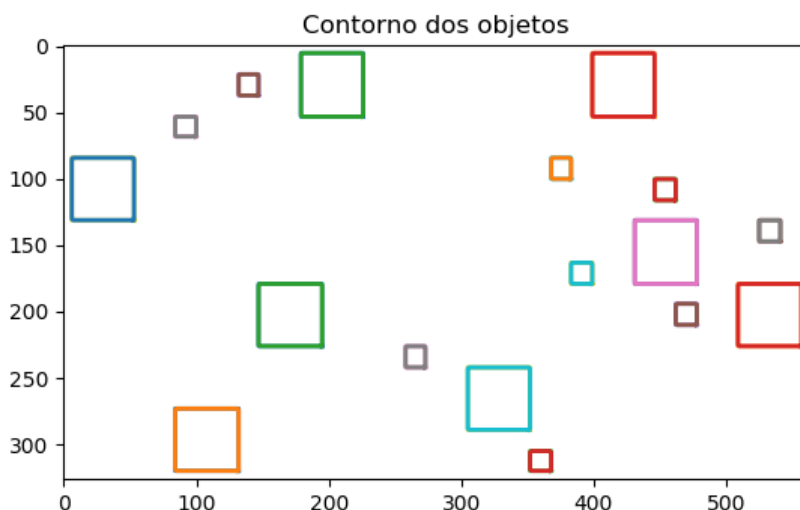
1. IMPLEMENTAÇÃO

A implementação apresentada neste trabalho foi feita na linguagem python em sua versão 3.6.3, e foram utilizadas as bibliotecas scypy, matplotlib, skimage e numpy para a manipulação e visualização dos dados.

Para realizar o carregamento da imagem que é dada como entrada foi utilizado o método `misc.imread()` que recebe o caminho da imagem que será carregada junto do parâmetro (`mode = "L"`) o que faz com que o objeto retornado seja uma matriz contendo os níveis de cinza de cada píxel. A Equação 1 exibe a transformação feita pela biblioteca para obter os níveis de cinza a partir da imagem RGB original.

$$L = R * \frac{299}{1000} + G * \frac{587}{1000} + B * \frac{114}{1000} \quad (1)$$

Tendo sido carregada a imagem em níveis de cinza, desejamos obter o contorno dos objetos que estão contidos na imagem, para isso foi utilizada a biblioteca `skimage` e o método `measure.find_contours()` que retorna uma lista contendo o conjunto de pontos que formam o contorno de cada objeto na imagem, esses foram os pontos utilizados para gerar o contorno dos objetos contidos na imagem. O resultado obtido por ser visto na Figura 1.



Após ter sido gerada a imagem contendo os contornos dos objetos, foi utilizado o método `measure.label()` para obter os rótulos dos objetos da imagem e o método `measure.regionprops()` que recebe o retorno de `measure.label()` como entrada e retorna uma lista contendo os objetos e seus atributos calculados, como área, perímetro e centroide.

A lista contendo os objetos e seus atributos foi utilizada para gerar a imagem informando os rótulos dos objetos a partir de seu centroide e para apresentar na tela seus

valores de área e perímetro, onde o atributo área foi também utilizado para gerar os valores do histograma apresentado.

2. EXECUÇÃO

Para que o código seja executado de forma correta basta executar o seguinte comando:

```
python3 T1.py caminho_da_imagem
```

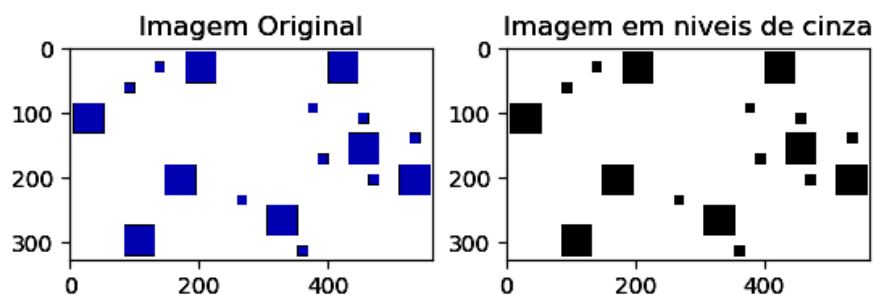
A saída será dada da seguinte forma:

1. Será exibido no console informações dos objetos contidos na imagem:
 - (a) área
 - (b) perímetro
2. Será exibido em diferentes janelas:
 - (a) imagem original
 - (b) uma imagem com o contorno de cada objeto contido nela
 - (c) uma imagem com os objetos enumerados
 - (d) histograma das áreas dos objetos

3. RESULTADOS

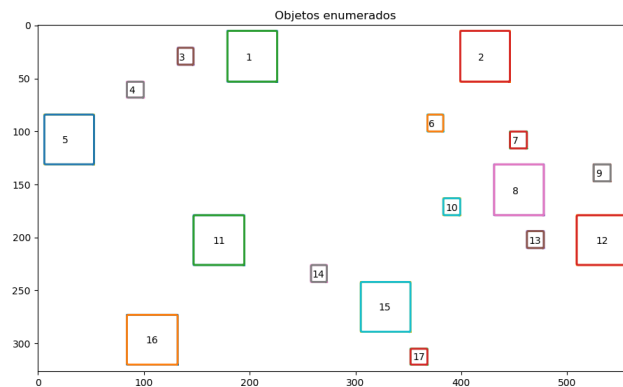
Nesta seção são apresentados os resultados obtidos passo-a-passo e uma breve discussão sobre a variação de parâmetros utilizados.

Ao carregarmos a imagem de entrada e gerarmos a mesma em níveis de cinza o resultado obtido é apresentado na Figura 3.



Durante os testes para a obtenção dos labels da imagem "objetos1.png", foi observado que independente da vizinhança utilizada, seja do tipo 4 ou 8 os resultados obtidos pelo método que recebia como entrada os labels para cálculo dos atributos desejados não apresentaram variação nos valores obtidos, o mesmo foi observado quando os testes foram realizados na imagem "objetos2.png".

Na Figura 3 é apresentada uma enumeração dos objetos da imagem, que foram colocados na imagem, se utilizando dos centroides de cada um dos objetos, um pequeno



ajuste de coordenadas teve que ser feito para que os rótulos melhor se alinhassem a seus objetos.

Um outro teste foi realizado para obter os atributos dos objetos da imagem, dessa vez fazendo uso da biblioteca `scipy.ndimage` para gerar os labels. Os resultados obtidos por essa mudança não apresentaram divergência dos obtidos pelo método `label` da biblioteca `skimage.measure()`.

Por último a Figura 3 exibe o histograma separando as classes de objetos encontradas na imagem em pequeno, médio e grande.

